2-6

SINC-LINK

TIMEX-SINCLAIR USERS CLUB

NEWSLETTER

Toronto, Ontario

PRINTED IN CANADA

B6146

NOVEMBER 1984 VOL 2/NO.6

EXECUTIVE OFFICERS

PRESIDENT: *George Chambers*
LIBRARIAN: *Martin Mauk*
ASS'T LIBRARIANS: *Y. Franken,*
D. Cooper,
T. Haller
TREASURER: *Greg Lloyd*
NEWS EDITORS: *Stan Piotrowski,*
John Roach
ACTIVITY DIRECTOR: *Ian Roberts*
LIASON OFFICER (Out-of-town members): *Chris Hart*
Mike Mitchell

LETTER FROM THE PRESIDENT

It is just a year since our club commenced operating with a little more formal structure to it. Overall, we have had a successful year. Two areas in particular have been outstanding, i.e. our newsletter has been published on a regular basis and with substantial content; and we have had a good series of hardware-software demonstrations at our meetings.

We have planned some improvements to our newsletter, namely two or three member-columnists, who will cover specific areas of interest i.e.: telecommunications; 2068; new products.

Improvements are also being made to our program library and in servicing our out-of-town members needs. These two aspects consume a lot of time and we are attempting to streamline this operation.

Your executive will be considering other ways of improving club operation. However, often these proposals entail more work, therfore, they can only be implemented on the basis of having volunteers come forward. This has not been a problem to date, but we invite anyone to come forward and volunteer your services.

Keeping the previous comment in mind we would welcome your comments-suggestions on how the club could better serve member needs. Out-of-town members may write to me at 14 Richome Court, Scarborough, Ontario, M1K 2Y1

Incidentally, at our last meeting 15 out of the 30 members present stated that they had the TS2068 computer. A whole new world is opening for our club!

Yours in Computing
George Chambers
(President)

STORE BASIC ON EPROM WITH AN EPROM PROGRAMMER
by Virgil Roman

Now you can store all your Basic or Machine Code programs on an Eprom chip and read them back in an instant. If you need a piece of hardware that can program the 2716, 2732 and 2764 Eproms compatible with all ZX-81/Timex computers, the "Programmer" is just what I recommend because it is very cost efficient.

Before I describe the system, let me say a few words about its simplicity; there is nothing complicated about it! The schematics have been improved by Rompak. The board design and artwork have been done by a small electronic company here in Toronto. The circuit is very simple and works very well. It comes completely assembled and tested, or in kit form for people with electronic background.

Eproms are solid state non-volatile (does not lose memory when computer switched of) circuit memory devices. They can be programmed by applying the necessary voltages and enable signals. It will not lose the stored data when the computer is turned off. They can be erased and then reprogrammed by exposing them by exposing them to ultra-violet light which will set all switches to one.

The Eprom Programmer plugs into the back of the computer and can be accessed with POKE commands or Machine Code. A 25 volt regulated power supply is required for 2716 and 2732 Eproms and a 21 volt power supply is required the 2764 Eprom. A "read board" is mapped to the unused 8-16K area for retrieving the programs stored on Eprom is also necessary (like the Hunter board).

Operating The "Programmer"

Plug the Eprom (2764) chip into the 28-pin socket, and note the polarity. If you are using an Eprom with 24 pins (2716 or 2732), pin #1 of the Eprom should go into pin #3 of the 28-pin socket (i.e. down 2 positions). Turn the computer on and make sure the "K" cursor appears. If it does, you are ready to "burn" an Eprom. Turn the external power supply to the appropraite voltage. (i.e. 21 or 25 volts).

Memory Allocations

The Eprom burner operates above Ramtop starting at address 32768: i.e. POKE 32768,205 will burn 205 into the first address of the Eprom, which can be read back into a READ board (Hunter, Rompak, etc.) at address 8192. The new Ramtop will be:

```
POKE 16388,254
POKE 16389,116
NEW
ENTER
```

This puts the new Ramtop at 29950 (74FE Hex).

To burn a Basic program on Eprom, we need a Machine Code routine to take the Basic program from the computer's memory and put ("burn") it into the Eprom. This M/L routine is called "Burning Routine". Once the Basic program is safe on the Eprom, we need a Machine Code routine, when called by a RAND USR command (i.e. RAND USR 8192), to stuff it back into RAM memory and run it. This machine code routine is called the "Booter Routine". Both routines are provided in the instruction sheets and should be POKEd above the new RAMTOP.

Circuit Description

There are two separate circuit sections. Refer to the schematic. The first section is memory write, the second circuit will decode the pulse, invert the signal and time enable the pulse to 50 ms. maximum all which is necessary for burning an Eprom.

Component Level Description

When a memory write is performed, the 74LS138 chip will decode this. The address range of 32768 - 40959 is outside of your 16K RAM pack and will not interfere with the normal function of your ZX/Timex computer. This pulse is inverted with a 74LS00 chip to clock the address and data into the 74LS273 chips. The signal also is used to trigger the 555 timer chip which will "burn" the Eprom with data and addresses stored in the 74LS273 chips. A 390K resistor and .1 uf capacitor controls the length of the timing pulse.

Recommendations

After using the "Programmer" for awhile, I have found it to be most useful. As a matter of fact, I consider it to be the most useful piece of hardware I have ever bought besides the computer itself. I strongly recommend it. The "Programmer" follows the programming specifications of INTEL memory devices and will be compatible with most Eproms on the market.

For more details on Eproms, read the article, "*COMPATIBILITY BETWEEN EPROMS*" by H. Pun - *Electronics Today (October 1984 - page 44)*.

If you deide to purchase the "Programmer", by all means, go ahead and order it and you will be glad you did:

Integrated Data Systems: (416) 466-5571
D & D Electronics Engineering Design: (416) 491-7493

SYNCBITS
by Ian Robertson

This column will, hopefully, be a regular item for our future newsletters. It is our intent to try and keep you up to date on what is happening in the world of TIMEX SINCLAIR computers, i.e., ZX-81, TS1000, TS1500, TS2068 and the QL. Maybe even an item, now and then, on the Spectrum.

Resources
ZX81/TS1000:

There are still some local vendors to draw on. For example; Integrated Data Systems- operated by fellow club member Ian Singer, Gladstone Electronics, Software Shop and from the publishers of Electronics Today or Computers Now magazines. A large source of Timex products is Timex themselves in Markham, on Hood Ave., but you must shop in person at their Warranty Service store located at their plant. If you phone they will simply tell you that Timex is no longer in the computer business. The price list is; TS1000 $40.00, TS1016 RAM 16K $25.00,TS2040 printer $80.00 and all software at $5.00 each. Gladstone have reduced their TS1000 price to $29.95.

TS2068

Again there are a couple of local sources; Integrated Data Systems and Gladstone Electronics. Gladstone have just reduced their price for the 2068 to $179.95, as well they have the Timex 2068 Technical Manual for $34.95.

Sams, the Indianapolis technical book publisher now has the Timex Sinclair 2068 Intermediate/Advanced Guide available at $9.95 US. I purchased my copy from E. Arthur Brown and the book seems to compliment the Timex Technical Manual.

QL

Constant monitoring of the U.K. computer press indicates that all is not well with the current model of the QL. Everyone agrees that while the QL is a great idea, the implimentation of this idea is less than great. For example- "superbasic" is far from super, in fact ot's reputed to be slow and tedious - QSDOS needs a lot of work to meet industry standards and the microdrives do not load and save with the integrity required in business or serious use. The biggest complaint is with the size of memory, apparantly 128K is not large enough to utilize the much vaunted "multi-tasking" aspect of the 68000. ther are even complaints about the quality of the PSION software packaged with the QL. Hopefully, all of this will be corrected prior to the QL entering the North American market by Sinclair Research in Boston via mail order, next Spring

Hardware/Software:

The big item in the news lately seems to be the number of "spectrum Compatible" ROM devices being sold. I have ordered the Russell Electronics " ROMSWITCH" from E. Arthur Brown and look forward to being able to use some of those 5,ØØØ or so Spectrum programs that are available in the U.K. There is also a U.K. supplier of the Spectrum ROM, which we will report on when the club receives our goods ordered.

Apparantly the 2Ø68 could be revived, as rumours persist that several parties are interested in either manufacturing or distributing this product. Only time (definately NOT Timex) will tell.

I have learned from Westridge Communications that their new version of the 2Ø5Ø modem software is issued ffor distribution and that they will not be producing the 2Ø68 "expansion box". In fact they say that they never even heard of this item before. Rumour laid to rest!

Magazines:

If you are looking for newstand magazines for our computers there are only 2 available, and they are both from the U.K.: "Your Computer", published monthly and "ZX Computing", published bi-monthly. Both are available from Litchmans bookstores in the Toronto area.

If you want really good TS1ØØØ and 2Ø68 U.S. dedicated magazines there are only 3: "Syntax" published monthly by The Harvard Group, Bolton Road, Harvard, Mass. Ø1451, at an incredible price of $48.ØØ U.S. per year. "Syncware News" published monthly by Thomas B. Woods (of ZX PRO/FILE fame), P.O. Box 64, Jefferson, N.H., Ø3583, at a more reasonable $16.95 U.S. per year.

"T-S Horizons" published monthly by T-S Horizons, Subscription Dept., 2ØØ2 Summit Street, Portsmouth, OH., 45662, at $21.ØØ U.S. per year.

8K EEPROM BOARD
by John Roach

In this project we are going to build a non-volatile memory expansion board that doesn't require batteries or a separate PROM blaster card. Everything we need will be on one simple card. Your cost to buy one of the popular PROM cards along with a separate programmer would probably be at least $15Ø dollars. Our project, hopefully, should cost less than $5Ø.

The heart of the board is a Seeq Technology 8K x 8 EEPROM, a memory chip that can be both erased and written to with only 5 Vdc. Once the chip has been programmed and the power to the computer turned off, the memory is permanent - even if you remove the chip from the board. This chip is also the most expensive component on the board. In

single quantity the price is $35. However, if there is enough interest in building the project we can get a good quantity discount.

Circuit Operation

Our circuit is designed to let the 8K of memory reside in the empty 8K to 16K portion of the memory map. When addressing this area, address lines A15 and A14 will be low and line 13 high. The MREQ, memory request line is also low. Line A13 is first inverted through IC1a so that when IC2 sees all its inputs low, it puts its output low. This signal has 3 jobs to do. The first is to disable the ROM in the computer to prevent memory ghosting. This we do by inverting the low output of IC2 through IC1b putting a high on the ROMCS line on the I/O bus. The second job is to enable our memory chip IC4 by putting a low on its CE chip enable pin. The third job we'll skip untill we look at a memory write cycle. For now we'll assume we are going to do a memory read.

So far we've decoded the address, enabled the memory chip and if we're doing a read the RD line on the I/O bus should be low, it's directly connected to IC4's output enable pin, OE. At this point the byte at the memory location we've addressed is plopped out onto the I/O data lines DØ to D7 ending the read cycle. So far everything has been fairly simple, but now let's take a look at a write cycle.

A write to the EEPROM only requires 5 volts but in order to program the chip the write pulse has to be 1 millisecond long. This is a lot longer than the Z8Ø write pulse which is about 1Ø nanoseconds. Here's where IC3 comes into play as well as the third job required of IC2. When IC2 sees an address in the 8K to 16K area of memory its output goes low, as we saw above, the output is inverted through IC1b and is used to enable IC3 which forms part of a single shot pulse stretching circuit. When the write pulse from the I/O bus is applied to an enabled IC3, its output delivers a 1 ms. low pulse to the write enable pin of our memory chip and the byte on the I/O bus line DØ to D7 is implanted forever into the EEPROM, or untill you do another write cycle to the same address. One limitation fo the chip is that you can only do 1Ø,ØØØ write cycles to any one individual byte address.

One other requirement in writing to the chip is that you can only overwrite a byte that is all 1's. This is not a problem, if you refer to Listing 1 which is a simple program to POKE data to sequential addresses you will see that a check is made by PEEKing first and if the location does not contain all 1's we just stuff them in before we write in the data we want there. Typically it shouldn't take more than about 8Ø seconds to write 8Kbytes to our memory chip. In spite of all the timing requirements and only being able to overwrite a byte thats all 1's it's probably a good idea to include the write protect switch shown on the diagram.

As far as applications for the EEPROM board, it can be used to store machine code programs that will run right from addresses on the board as the read time is 25Ø ns. and the computer sees it as RAM. It can also store BASIC programs although they can't be run from the

board, you would simply down load the block of memory containing the tokenized program to the EEPROM and up load it back to the BASIC area in RAM when you want to run it.

In the concluding portion of this project, appearing in the next newsletter, we will cover construction details, PC board layouts, where to get parts, more software for loading and some sugestions for modifications to make the board usable on the 2068 - both on the expansion bus and the cartridge dock.

Program Listing 1

```
10 REM ** EEPROM LOADER **
20 REM S= STARTING ADDR. OF
30 REM    DATA IN RAM.
40 REM N= NUMBER OF BYTES -1.
50 REM E= EEPROM START ADDRS.
60 LET S= 16436
70 LET N= 2047
80 LET E= 8200
80 FOR I= D TO D+N
90 IF PEEK (E) = 255 THEN GOTO 130
100 POKE P, 255
110 FOR J= 1 TO 2
120 NEXT J
130 POKE P, PEEK (I)
140 NEXT I
150 PRINT "I'M LOADED, EH!"
160 STOP
```

PROGRAM CREDITS
by G.F. Chambers

In looking through our library of programs, I am struck by the number of contributions which contain no hint of their source or origin. They are something like a ship without a name or a letter without an address. This is unfortunate because, at the very least, someone who went to the trouble of creating the program or entering it does not get appropriate credit. It also makes it difficult for the next user to refer back to its origins should the need arise.

The introduction could be as simple as a two-line entry as follows:

```
10 REM GRAPHICS DEMONSTRATION
20 REM FROM ZX81 PROGRAMMING MANUAL CHAPTER 18, PAGE 121
```

An original program probably deserves that more credit be given the maker. Take a look at the partial listing shown in the example. Here, the program is an "Auto-Run"; that is, if it is SAVEd by a direct command - GOTO 9900, it will SAVE itself. Then, when you LOAD it into your computer, it will automatically RUN or continue at 9910. (If you wish to use RUN or GOTO to start a program, the initial line in the program should be: GOSUB 9910 to enter into this routine and line 9930 should read RETURN).

To get back to the original subject, note line: 9910 LIST 9970. This will cause the program to be listed automatically on the screen from line 9970 onwards. This will display any notice about the program you care to present.

Several points should be noted. The listing to be displayed needs to be at the end of the program and not to be so lengthy that it exceeds the screen size (21 lines). If the listing called up in line 9910 exceeds the screen size, it will cause a CODE 5 report and the program will stop.

Notice that lines 9997 & 9998 contain an instruction to the user to press a key. This is because although the listing appears on the screen, the computer is still in the program operation mode and in fact is at line 9920 PAUSE 4E4; i.e. waiting for a key to be pressed (waiting "forever" - 4E4 sounds like the word 'forever'). After a key is pressed, the program will proceed to the next line: 9930 GOTO 2000 which happens to be start of the main program in the particular program this example was taken from. (PAUSE 4E4 = PAUSE 4000 = PAUSE FOREVER. See Sinclair/Timex manual, Chapter 19).

PROGRAM EXAMPLE OF CREDITS

```
9900 SAVE "METRIC"
9910 LIST 9970
9920 PAUSE 4E4
9930 GOTO 2000
9970 REM
9971 REM
9972 REM      METRIC CONVERSION
9978 REM      -----------------
9979 REM
9980 REM
9982 REM     AN ORIGINAL PROGRAM
9983 REM
9984 REM         WRITTEN  BY
9985 REM        G.F. CHAMBERS
9988 REM          (C) 1984
9989 REM
9991 REM
9992 REM        DATA TAKEN FROM
9993 REM       THE BOOK ENTITLED
9994 REM     "COMPLETE HANDYMAN'S
9995 REM         ENCYCLOPEDIA"

9997 REM       TO START PROGRAM
9998 REM         PRESS ANY KEY
9999 REM
```

TS 2068 PROGRAMMING
by John Roach

Recognizing UDG's

As you know from the manual, there are 21 User Defineable Graphic characters which can be assigned from CODE 144 to CODE 164. To show what you are doing and to keep program LISTing as clear as possible, it's best to use the CHR$ function rather than shifting to the Graphics mode.

```
10 PRINT CHR$ 144; CHR$ 145; CHR$ 146
```

It's instantly recognizable as UDG's.

UDG Loader

This routine is not in the manual but it saves a lot of typing time not using all the BIN values.

```
9000 FOR g=144 TO 164          (or less)
9010 FOR n=0 TO 7
9020 READ a : POKE USR CHR$ q + n,a
9030 NEXT n
9040 RESTORE : RETURN
9050 DATA.................
```

Each data statement contains 8 numbers per line; each number is the decimal value for the binary bits in UDG character row. Each data statement defines one character.

Fancy Titles

Assign a string to T$ less than or equal to 32 characters for the title and a value to the variable "pa" less than 22 for PRINT AT line number:

```
10 LET T$ = "HERE IS A PROGRAM TITLE"
20 LET pa = 10
30 GOSUB 100
40 STOP

100 LET M = LEN T$ : IF M/2 < > INT (M/2) THEN LET T$ = T$ + " "
: LET M = M +1
110 FOR N = 1 TO M/2 : PRINT AT pa, 16-N; T$ ( TO N); T$ (M - N +
TO  ): BEEP .02,2 * N : NEXT N : RETURN
```

MACHINE CODE PROGRAMMING

The following is a list of the keyboard values when K-SCAN called in M/L (CALL 699 or in code: 205, 187, 2) which returns the value in the HL register. Note, these are all computed values so to find the value in 'H' & 'L' we must find it ourselves. e.g., the value returned to HL when the number '5' is pressed (or left arrow) is 57335. Therefore, to find the value contained in 'HL',:

```
10 LET H = INT (57335/256)      :223
20 LET L = 57335 - (H * 256)    :247
30 PRINT "H=";H, "L=";L
40 STOP
```

Therefore, if we were testing to see if the user was pressing the '5' key (or left arrow), we would first CALL 699 then LD B,223 & LD C,247; LD A,H & CP B; LD A,L & CP C. If they matched, then continue on with the program.

Whenever you're programming in machine code, do not forget that there are many routines in ROM that we can use (which will be discussed as we go along). Also, when reading articles about machine code, they sometimes have a useful routine which can be called. For example, in Toni Baker's book, "MASTERING MACHINE CODE ON YOUR ZX-81", has many useful subroutines. We will be using some of them such as printing numbers to the screen for increasing/decreasing the scores; or on page 59 is a very useful routine for printing a complete picture or text to the screen in machine code.

TRIVIA:

250 baud means about 25 bytes per second; 1500 baud means about 150 bytes per second: for LOADing or SAVEing values which is why the ZX-81 takes so long to LOAD or SAVE!

BASIC PROGRAMMING

Side Scroll a Message

```
100 LET A$ = "print your message with as many characters as you
like but not less than 33, then 5 spaces then an asterisk (*)"
110 LET N = 1
120 IF A$ (N + 31) = "*" THEN GOTO 110
130 PRINT AT 10,0; A$ (N TO N + 30)
140 FOR I = 1 TO 3
150 NEXT I
160 LET N = N + 1
170 GOTO 120
```

LETTER FROM THE NEWS EDITORS

Well that's it again for this Newsletter. As you can see, with more submissions, we can print a varied type of Newsletter that should appeal to the various interests of club members.

When submitting articles, please use double spacing when typing and write neatly of it is a written submission. All articles have to be re-typed into the Newsletter and it is much easier for us. Also, we don't have an Art Department so any drawings, etc. will be entered as submitted; make them neat and on a separate piece of paper from your article.

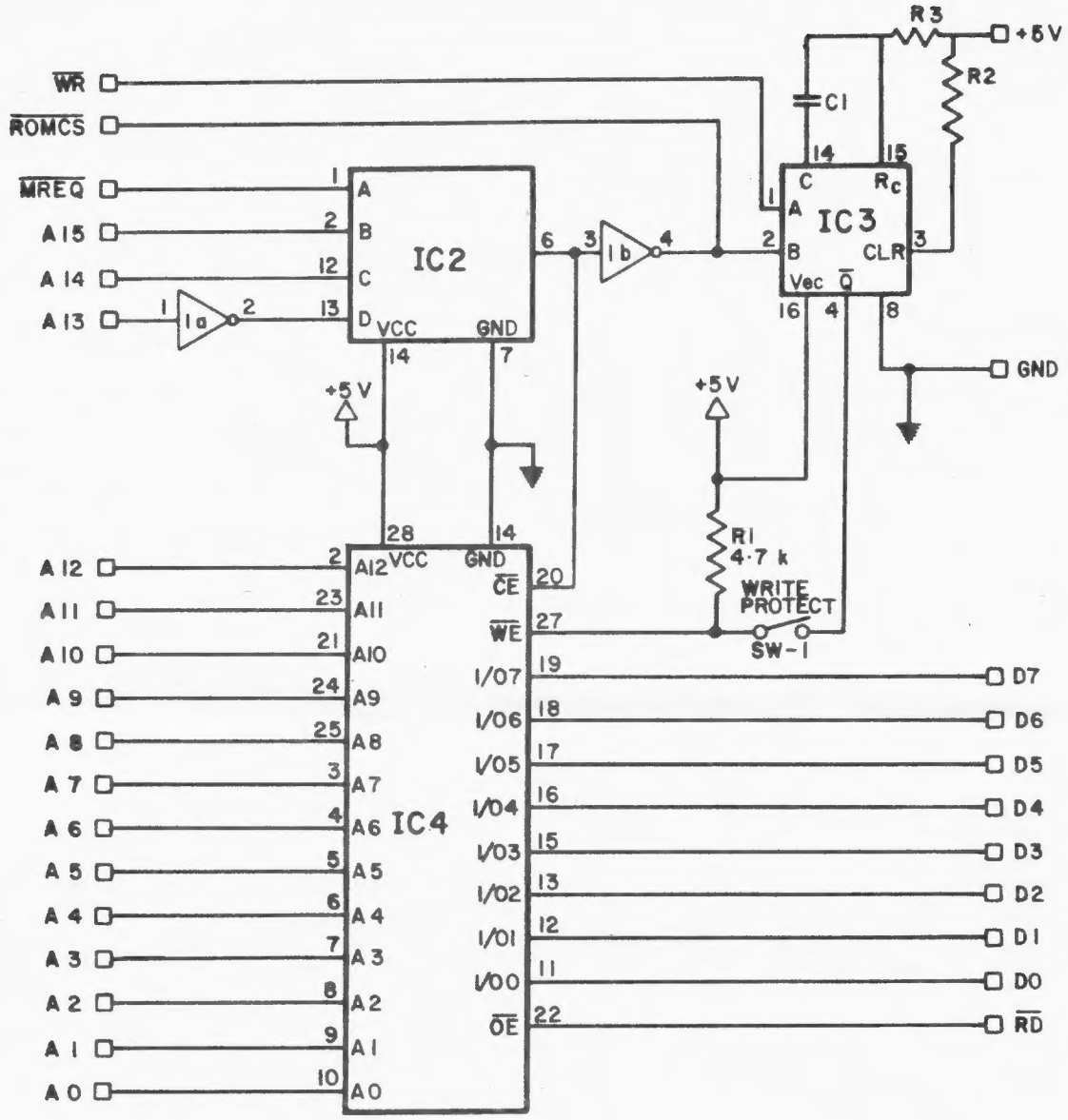
So until the next Newsletter (in January), have fun programming.

Stan & John
News Editors

Table of 'Key Values'

Key	Hex. Value	Dec. Value	Key	Hex. Value	Dec. Value
1	FDF7	65015	EDIT	FCF7	64759
2	FBF7	64503	AND	FAF7	64247
3	F7F7	63479	THEN	F6F7	63223
4	EFF7	61431	TO	EEF7	61175
5	DFF7	57335	←	DEF7	57079
6	DFEF	57327	↓	DEEF	57071
7	EFEF	61423	↑	EEEF	61167
8	F7EF	63471	→	F6EF	63215
9	FBEF	64495	GRAPHICS	FAEF	64239
0	FDEF	65007	RUBOUT	FCEF	64751
Q	FDFB	65019	‖	FCFB	64763
W	FBFB	64507	OR	FAFB	64251
E	F7FB	63438	STEP	F6FB	63227
R	EFFB	61435	<=	EEFB	61179
T	DFFB	57339	<>	DEFB	57083
Y	DFDF	57311	>=	DEDF	57055
U	EFDF	61407	S	EEDF	61151
I	F7DF	63455	(	F6DF	63199
O	FBDF	64479	)	FADF	64223
P	FDDF	64991	‖	FCDF	64735
A	FDFD	65021	STOP	FCFD	64765
S	FBFD	64509	LPRINT	FAFD	64253
D	F7FD	63485	SLOW	F6FD	63229
F	EFFD	61437	FAST	EEFD	61181
G	DFFD	57341	LLIST	DEFD	57085
H	DFBF	57279	"	DEBF	57023
J	EFBF	61375	–	EEBF	61119
K	F7BF	63423	+	F6BF	63167
L	FBBF	64447	=	FABF	64191
NEWLINE	FDBF	64959	FUNCTION	FCBF	64959
Z	FBFE	64510	:	FAFE	64254
X	F7FE	63486	;	F6FE	63230
C	EFFE	61438	?	EEFE	61182
V	DFFE	57342	/	DEFE	57086
B	DF7F	57215	*	DE7F	56959
N	EF7F	61311	<	EE7F	61055
M	F77F	63359	>	F67F	63103
•	FB7F	64383	.	FA7F	64127
SPACE	FD7F	64895	£	FC7F	64639
NO KEY	FFFF	65535			

8K EEPROM BOARD



IC1 - 74LS04 HEX INVERTER
IC2 - 74LS32 QUAD OR GATE
IC3 - 74LS123 MONOSTABLE MULTIVIBRATOR
IC4 - 52B33H 8K x 8 EEPROM